

Ce sujet comporte 2 pages. Tous les documents sont autorisés. Le barème est indicatif. Toutes les réponses doivent être justifiées (de façon concise mais précise).

Partie I - Arbre de recherche (7 points)

Soit le programme

$a([], N) :- p(N), !.$
 $a([X|L], N) :- b(X), !, c(L,M), N \text{ is } M+1.$
 $a([X|L], N) :- d(L,N).$

$b(X) :- e(X,Y), !, f(Y).$
 $b(X) :- f(X).$

$e(X, "Val1") :- 10 \text{ is } X+3.$
 $e(X, "Val2") :- X < 18.$
 $e(X, "Val3") :- X > 37.$

$f([X|L] :- 0'a=< X, X<0'e.$

$d([X|L], N) :- d(L, M), N \text{ is } M+1.$
 $d([], 0).$

Question 1

Dessiner l'arbre de recherche du but suivant : $a([12, 28, 13], K).$

Indiquer les témoins des preuves si il y en a. Indiquer clairement pour chaque coupure les branches qu'elle coupe.

Partie II - Programmation logique (13 points)

Un atelier emploie 4 ouvriers, O_1, O_2, O_3 et O_4 , et fabrique 4 produits, P_1, P_2, P_3 et P_4 . Les ouvriers ont des compétences différentes et ne fabriquent pas tous les produits avec le même profit. L'ouvrier O_1 produit P_1, P_2, P_3 et P_4 avec les profits 7, 1, 3 et 4, l'ouvrier O_2 avec les profits 8, 2, 5 et 1, l'ouvrier O_3 avec les profits 4, 3, 7 et 2, et l'ouvrier O_4 avec les profits 3, 1, 6 et 3. Un ouvrier ne peut produire qu'un produit à la fois. On désire confier aux ouvriers la fabrication des produits en assurant un profit total supérieur à une limite inférieure de 19.

Le programme suivant, utilisant la librairie `clpfd`, résoud ce problème. Chaque ouvrier se voit attribuer l'une des 4 tâches, de la façon suivante :

```
?- domain( [O1, O2, O3, O4], 1, 4) ,
   all_different( [O1, O2, O3, O4] ) ,
   element( O1, [7, 1, 3, 4], Profit1 ) , % l 'élément de rang O1 de la liste est Profit1
   element( O2, [8, 2, 5, 1], Profit2 ) ,
   element( O3, [4, 3, 7, 2], Profit3 ) ,
   element( O4, [3, 1, 6, 3], Profit4 ) ,
   ProfitTotal #= Profit1 + Profit2 + Profit3 + Profit4 ,
   ProfitTotal #>= 19 ,
   labeling( [], [O1, O2, O3, O4] ).
```

Une des réponses est [1, 2, 3, 4] pour un profit total de 19.

On appelle ce programme le «programme naïf». On souhaite généraliser ce programme de telle sorte qu'il consulte les paramètres du problème à résoudre dans une base de données. Celle-ci comprend une relation

`produit(Ouvrier, Produit, Profit)` - par exemple `produit('Arthur', lamelle, 7)`.

et une relation

`objectif(ProfitMinimal)` - par exemple `objectif(19)` .

On suppose dans cette partie qu'il y a toujours autant d'ouvriers que de produits.

Question 3 (2 points)

À quoi correspond la liste [O1, O2, O3, O4] ? Ecrire la requête permettant de former cette liste dans le cas général.

Question 5 (3 points)

On va devoir associer un numéro à chaque produit. Écrire la requête permettant de récupérer la liste des produits, puis de lui associer une liste de numéros.

Question 6 (5 points)

En utilisant les éléments de programme précédents, donner une variante générique du programme qui affiche ses réponses sous la forme

"L'ouvrier O fabrique le produit P".

Question 7 (3 points)

On souhaite prendre en compte un objectif de quantité de produits ; le produit P_i doit être fabriqué en quantité Q_i . Dans cette variante du problème les ouvriers produisent tous la même quantité totale, mais la production de chacun est ventilée sur les produits de façon à obtenir un profit total supérieur à une limite donnée. Par exemple, l'ouvrier O_1 fabriquera Q_{11} produits P_1 , Q_{12} produits P_2 , etc. Son profit sera alors $Q_{11} \times 7 + Q_{12} \times 1 + \dots$

Donner le programme naïf qui résoud ce problème dans le cas 4 ouvriers et 4 produits.